CLAIMS

What is claimed is:

1. A system for performing code optimization, comprising:

an optimizing analyzer within a compiler to generate a first optimizing transformation and a second optimizing transformation and their satisfying conditions for a compiled code; and

an optimization transformation module within a linker to determine which of the first and second optimizing transformations should be selected when the compiled code is linked with other compiled codes, and to execute the selected one of the first and second optimizing transformations at link-time.

2. The system of claim 1, wherein the optimization transformation module determines which of the first and second optimizing transformations should be selected by checking the satisfying conditions with information only available at link-time.

3. The system of claim 1, wherein the first and second optimizing transformations and their satisfying conditions are in the form of conditional relocation operations.

4. The system of claim 1, wherein the first and second optimizing transformations are address base binding optimizing transformations.

5. The system of claim 1, wherein the first and second optimizing transformations are function cloning optimizing transformations.

6. The system of claim 1, wherein the first and second optimizing transformations are data allocation optimizing transformations.

7. A compilation environment, comprising:

a compiler to compile a source code into a compiled code;

a linker to link the compiled code with other compiled codes into an executable program;

an optimizing analyzer within the compiler to generate a first optimizing transformation and a second optimizing transformation and their satisfying conditions for the compiled code; and

an optimization transformation module within the linker to determine which of the first and second optimizing transformations should be selected when the compiled code is linked with the other compiled codes, and to execute the selected one of the first and second optimizing transformations at link-time.

8. The compilation environment of claim 7, wherein the optimization transformation module determines which of the first and second optimizing transformations should be selected by checking the satisfying conditions with information only available at link-time.

9. The compilation environment of claim 7, wherein the first and second optimizing transformations and their satisfying conditions are in the form of conditional relocation operations.

10. The compilation environment of claim 7, wherein the first and second optimizing transformations are selected from a group comprising address base binding optimizing transformations, function cloning optimizing transformations, and data allocation optimizing transformations.

11. A method of performing code optimization, comprising

generating a first optimizing transformation and a second optimizing transformation and their satisfying conditions for a compiled code at compile-time;

determining which of the first and second optimizing transformations should be selected when the compiled code is linked with other compiled codes; and

executing the selected one of the first and second optimizing transformations at link-time.

12. The method of claim 11, wherein the first and second optimizing transformations and their satisfying conditions are generated in the form of conditional relocations.

13. The method of claim 11, wherein determining which of the first and second optimizing transformations should be selected further comprises checking the satisfying conditions with information only available at link-time to determine which of the first and second optimizing transformations should be selected.

14. The method of claim 11, wherein the first and second optimizing transformations are address base binding optimizing transformations.

15. The method of claim 11, wherein the first and second optimizing transformations are data allocation optimizing transformations.

16. The method of claim 11, wherein the first and second optimizing transformations are function cloning optimizing transformations.

17. An article of manufacture comprising a machine accessible medium including sequences of instructions, the sequences of instructions including instructions which, when executed, cause the machine to perform:

generating a first optimizing transformation and a second optimizing transformation and their satisfying conditions for a compiled code at compiler-time;

determining which of the first and second optimizing transformations should be selected when the compiled code is linked with other compiled codes; and

executing the selected one of the first and second optimizing transformations at link-time.

18. The article of manufacture of claim 17, wherein the first and second optimizing transformations and their satisfying conditions are generated in the form of conditional relocations.

19. The article of manufacture of claim 17, wherein determining which of the first and second optimizing transformations should be selected further comprises checking the satisfying conditions with information only available at link-time to determine which of the first and second optimizing transformations should be selected.

20. The article of manufacture of claim 17, wherein the first and second optimizing transformations are address base binding optimizing transformations.

21. The article of manufacture of claim 17, wherein the first and second optimizing transformations are function cloning optimizing transformations.

22. The article of manufacture of claim 17, wherein the first and second optimizing transformations are data allocation optimizing transformations.